Replay Detection Based on Semi-automatic Logo Template Sequence Extraction in Sports Video

Zhonghua Dang[1] , Jun Du[2], Qingming Huang[1], Shuqiang Jiang[3]
Graduate School of Chinese Academy of Sciences, Beijing 100039, China[1]
NEC Laboratories, China, Beijing 100084, China[2]
Institute of Technology,Chinese,Bejing, 100080,China[3]
{zhdang, jdu,qmhuang,sqjiang}@jdl.ac.cn

Abstract

This paper proposes a novel approach to replay detection in sports video. The proposed method consists of efficient semi-automatic logo template sequence extraction and logo template sequence based replay detection. In the beginning, two video segments are marked manually. Then a logo template sequence extraction algorithm is adopted to extract logo template sequence automatically. Once the logo sequence is extracted, the sequence is used to locate the replay segments accurately by the template sequence matching method. The experiments are carried out on various types of sports videos. The experimental results show that our algorithm can extract logo template sequence accurately and detect replays effectively. Moreover, our algorithm is generic and easy to use.

1. Introduction

In sports video, replay usually highlights the semantic events. Replay detection is a technique to locate replays in sports video automatically and it is quite useful to extract semantic events in sports video analysis. Some early researchers have focused on the replay detection techniques in the past decade. Kobla et al. [1] detected replays for sports/non-sports video classification. Their method was block-based, and used motion vectors in the MPEG-1 domain. Pan et al. [4] utilized HMM to model replay segments by considering a zero-crossing measure for both the frequency and amplitude of the fluctuations of adjacent frame difference, while a single field inside a replay, however, must be pinpointed in advance. Later, they proposed replay detection based on replay transition [5]. They used the method [4] to detect two replay segments first, then, they searched two frames most similar at one of the two search areas preceding the two replay segments, and finally, a verification procedure was employed. Tong et al. [6] detected replays by using logo template matching. Their method firstly detected some logo transitions and further extracted logo samples from them at the beginning of their method, and then they extracted the logo template from these samples. Further, they employed this template to detect other logos. After all logos were obtained, the video was partitioned into segments with taking logo as boundaries. Finally, an SVM classifier was used to identify replay by using shot and motion features of one segment.

In this paper, we propose a novel and robust replay detection method. The method consists of two primary steps. The first step is to extract logo template sequence, and the second is to locate the replay segments by using the logo template sequence match method. It is well known that in most sports videos there exists a special transition at both start and end of a replay, in which a highlighted logo comes in and out gradually. We call this transition as "logo transition". Our motivation is to use the logo transition positions to locate replay segments. As long as the positions of logo transitions are located, the replay segments can be detected easily. Based on our knowledge, there are four difficulties in locating the logo transitions: a) the time of logo transition is random, b) the position of logo in frame is random, c) the shape of logo is random and changing, and d) the texture and color of logo is random. To overcome these difficulties as much as possible, we propose a semi-automatic logo template sequence extraction algorithm based on the

This work is supported by National High-Tech R&D Program (863 Program) under the grant No. 2006AA01Z117 and NEC Labs, China.

0-7695-2929-1/07 $25.00 © 2007 IEEE
DOI 10.1109/ICIG.2007.73

unsupervised frame segmentation of color-texture regions. Firstly, the algorithm is employed to extract the logo template sequence from video segment containing part or full of logo transition accurately. The logo template sequence is used to locate the positions of logo transition by logo template sequence matching. Then the positions of logo transition are employed to detect replay segments in sports video. The framework of our algorithm is shown in Fig. 1.

The rest of this paper is organized as follow. In Section 2, logo template sequence extraction is discussed. In Section 3, we describe the logo transitions detection and the replay segments location. In Section 4, the experimental results are given.

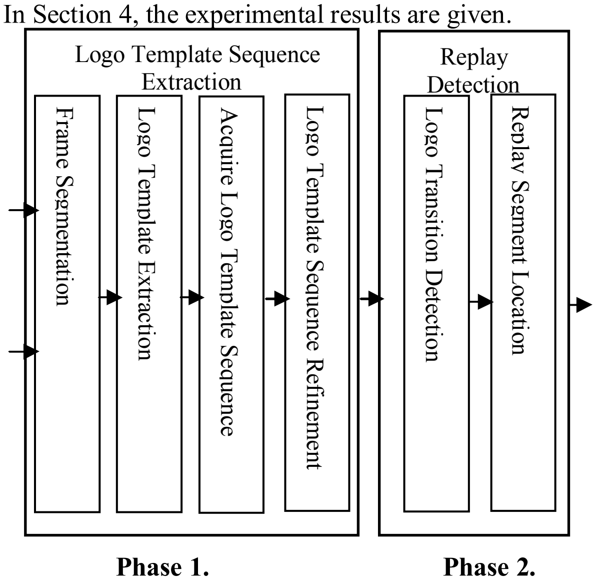

Fig.1.The flowchart of the proposed algorithm.

2. Logo Template Sequence Extraction

Logo template sequence consists of logo templates, thus, logo template sequence extraction is completed by extracting logo template from each frame in one video segment.

Generally, a logo consists of one or several color regions in one frame. Therefore, logo template can be extracted by merging logo color regions in one frame. And then, logo template sequence is extracted by extracting logo template from each frame in video segment. Based on such observation, a color region based logo template sequence extraction algorithm is proposed. Firstly, in one video segment, each frame is segmented into several logo and non-logo color regions. Secondly, logo color regions are acquired by removing non-logo color regions from each frame. Logo template is extracted by merging the logo color regions in each frame. In this way, all logo templates are extracted from each frame in one video segment. Logo template sequence can be formed by these logo templates. We extract two logo template sequences from two video segments. A precise logo template sequence (*TS*) is acquired by making a further match one by one between the two sequences.

The proposed logo template sequence extraction algorithm uses and processes two input video segments that contain full or part of logo transition. These two segments are marked manually. The two video segments are named as *C1* and *C2* respectively. In this paper, we assume that *C1* contains two frames and *C2* contains one frame at least. The following subsections are detailed description on the proposed logo template sequence extraction algorithm.

2.1 Frame Segmentation

Video frame consists of several color regions in which all pixels are most similar. To extract logo color regions from a frame, the frame is segmented into color regions at first. The JSEG algorithm [9] is adapted to segment video frames. JSEG algorithm is an unsupervised segmentation algorithm and works well. After segmentation, the frame is segmented into several logo and non-logo color regions. Fig. 2 shows color region images of one frame. In each region image, the black pixels are background pixels not belonging to the region. In Fig. 2, the first is the segmented frame image. Only the second region image is the logo region. The others are non-logo color regions. The logo regions are extracted with the method described in the next subsection.

Fig. 2. After frame segmentation, color region images in the frame.

2.2 Logo Template Extraction

Logo color regions are extracted by removing non-logo regions from frame, and then logo color regions are merged to form logo template in frame.

In the paper, non-logo region is named as *BKR*. During the period of logo transition, *BKRs* are minor movement with respect to their size. In contrast to *BKRs*, logo regions are major movement. The movement discrepancy of *BKRs* and logo regions is employed to remove *BKRs*. The following two features are used to describe the movement discrepancy and remove *BKRs*. a) Color_Dist: it is calculated as the sum of absolute difference of *RGB* components in two color regions. b) Overlap_Ratio: it is calculated as the ratio of the number of pixels with the same coordinates in two regions to the average number of pixels in the same two regions.

The feature Color_Dist is used to track color region between adjacent frames in video. If two color regions in adjacent frames have similar color, they are considered to be the same color region. The more similar two region colors are, the smaller their Color_Dist is. Region *A* and region *B* are contained in the current and subsequent frame respectively, and they are considered to be the same color region, if their Color_Dist is not over an adaptive threshold *T*. *T* is calculated as *256/M*, where *M* is the number of color regions in the current frame. If region *A* and region *B* are the same region in adjacent frames, region *B* is considered to be the matching region of region *A*.

The feature Overlap_Ratio is used to express color region motion degree between adjacent frames in video. The bigger the color region motion degree is, the smaller the corresponding Overlap_Ratio is. The smaller the color region motion degree is, the bigger the corresponding Overlap_Ratio is. Logo color region motion degree is bigger, thus, its Overlap_Ratio is smaller. *BKR* motion degree is smaller, thus, its Overlap_Ratio is bigger. If Overlap_Ratio of two color regions is over an adaptive threshold *V*, the two color regions are marked as *BKRs*. In Fig. 3, the dashed and the solid curves illustrate the Overlap_Ratio values of *BKRs* and logo regions in one video segment respectively. Overlap_Ratio values for *BKRs* and logo regions are distinct. Thus the Overlap_Ratio values for *BKRs* and logo regions are classified into two classes by *K-Mean* clustering algorithm [10]. *V* is set as the bigger clustering center.

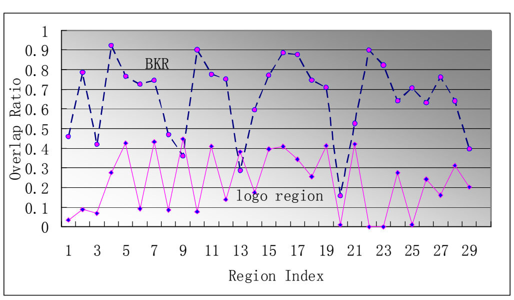

Fig.3. Logo regions and BKRs Overlap_Ratio.

The algorithm of removing *BKR* from one video segment is described as follow.

1. For each region *A* in one frame, find its matching region *B* in the next frame. If its matching region is not found, the region *A* is marked as *BKR*.
2. Mark region *A* and region *B* as *BKR*, if their Overlap_Ratio is over threshold *V*.
3. Carry out step 1 and 2 on each frame in one video segment to find *BKRs*. These *BKRs* form a set *SBKR*.
4. For each region in the video segment, if its matching region is found in *SBKR*, the region is marked as *BKR* and merged into *SBKR*. Repeat this step until *SBKR* is not expanding.

These steps are stepwise refinement process. After steps 1, 2 and 3, the *BKRs* with Overlap_Ratio is over *V* are marked. In step 4, an iterative process finds the remaining *BKRs* with Overlap_Ratio not more than *V*. After these steps, in one video segment, there are only color logo regions in each frame. Logo template is extracted by merging remaining logo color regions in each frame.

2.3 Acquire Logo Template Sequence

Logo template sequence is extracted by extracting logo template from each frame with the method mentioned in subsection 2.2 in one video segment. In this way, the first logo template sequence is extracted from *C1*.

In order to refine logo template sequence in the next subsection, *C1* and *C2* need contain the same part of logo transition. But when *C1* and *C2* are marked, their parts of logo transition may be not identical. Thus, *C2* is extended forward and backward respectively to include the whole logo transition. The video segment *C3* whose part of logo transition is identical to that in *C1* is searched from the extended *C2*. From the start frame of *C2*, *C3* is searched according to Equation (1)

$$S = \frac{1}{L}\sum_{i=1}^{L} d(T_i, I_{k+i}), \qquad (1)$$

where L is the length of the first logo template sequence and k is the start searching frame. $d(T_i, I_{k+i})$ is used to calculate histogram similarity between logo template and frame with histogram intersection method, where T_i and I_{k+i} belong to the first logo sequence and the extended *C2* respectively. *C3* is the video segment between frame *k* and frame *k+L-1*, if *S* reaches the peak value at frame *k*. The peak value of *S* is the optimal match value (*OMV*). The second logo template sequence is extracted from *C3*. Fig. 4 shows some example images in different logo template sequence. As showed in Fig.4, the first and the second logo template sequences are noisy, but they can be refined with the mentioned in the next subsection.

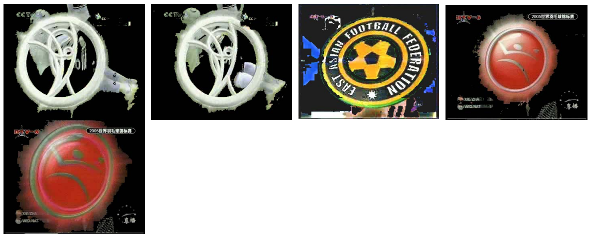

Fig. 4. Example images of temple in different logo template sequence.

2.4 Logo Template Sequence Refinement

The first and second logo templates are noisy, thus, they need to be refined. After the refinement process, a precise logo template sequence (*TS*) is acquired.

The first and the second logo templates have the same length and their templates align one by one. In one template in the first logo template sequence and its aligning template in the second template sequence, all pixels sharing the same spatial locations are extracted to form one template of *TS*. In this way, all templates of *TS* are extracted. *TS* is a precise and stable logo template sequence in the whole video. Fig. 5 shows some example logo template images in different *TS*. In contrast to Fig. 4, most of noisy regions are removed, and each template in *TS* is much more precise. *TS* is employed to detect replay segments with the method mentioned in the next section.

Fig.5.Example images of template in different *TS*.

3. Replay Detection

In sports video, two logo transitions often immediately occur and follow by replay segments. Based on such observation, logo transition positions can be automatically detected with logo template sequence match method via using *TS* first, and then, the logo transition positions are paired to locate the replay segments.

3.1 Logo Transition Detection

Once *TS* has been extracted, the other logo transitions can be detected via logo template sequence matching method, in which color histogram similarity is employed. The histogram similarity (*HS*) of *TS* and *L* continuous frames in video is calculated as Equation (1). The logo transition detection algorithm is described as follows:

1. Continuous Frames between current frame *N* and *N+L-1* are used to calculate *HS*.

2. Set *N = N+1*, if *HS* is over threshold value *R*, add it to an adaptive size buffer, and go to Step 1, otherwise if *HS* is not over *R* and the buffer is not empty, go to Step 3, or if *HS* is not over *R* and the buffer is empty, go to Step 1.

3. Select the maximum value of *HS* from the buffer where the index value of *N* corresponding to the maximum *HS* is the logo transition position. Reset the buffer and go to Step 1.In this algorithm, *R* is a threshold and evaluated according to the *OMV*.

3.2 Replay Segments Location

After all of the logo transitions are detected, they are used as the boundaries of relay segments. If the temporal duration of video segment between two adjacent logo transitions is in the range of (*TL, TH*), the video segment is detected as replay segment. *TL* and *TH* are empirically determined based on the fact that a replay usually does not last too long or too short. The proposed replays detection method can detect replay segments effectively and rapidly in sports video.

4. Experiments

The performance of the proposed method was evaluated on several MPEG-1format sports video. We collected these sports videos from TV. These sports videos contain different kinds of logo transitions, and the total size of the videos is about 10GB. In these videos, the replay segments are marked manually as ground truth to test the proposed replay detection method. The recall and precision are used to evaluate the method performance. The proposed method is controlled by two parameters: the *TS* length *L* and the threshold *R*. We test the algorithm under the various situations of the two parameters respectively. Two recall-precision curves are given to illustrate the influences of the two parameters in detail.

We test the method performance under various *TS* length. The longer *TS* can reduce noise more effectively during the process of logo transition detection with logo template sequence match method. Thus, the longer *TS* induce higher performance of the method. A typical logo transition lasts 0.5-0.8 seconds or 15-24 frames. Usually, the clearer logos locate the middle part of the logo transition. Thus, from the middle of logo transition, we cut five *TSs* with different length. The testing *TS* lengths are 3, 4, 5, 6, and 7. The performance of the proposed method under different length *TSs* are shown in Fig. 6 The method overall performance is higher with longer *TS*, but it only changes slightly when the *TS* length is over 7. In the experiment, *TS* length is 7 to acquire better performance.

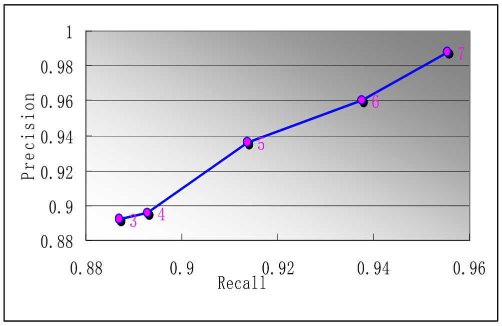

Fig. 6. The *P-R* curve of *TS* length *L*.

We test the method performance under various threshold *R* values. The lower *R* values induce lower precision and higher recall performance. In contrast, the higher *R* values induce higher precision and lower recall performance. The threshold *R* is evaluated according to the *OMV* value calculated in the subsection 2.3. The *R* is evaluated as 60%, 70%, 80%, 90%, 100% of *OMV* to test the method performance. Fig. 7 shows the method *P-R* curve with different *R* values. The higher *R* values miss more correct logo transitions, the lower *R* values detect more incorrect logo transitions. In the experiment, *R* is evaluated as 70% of *OMV* to obtain overall better precision and recall performance. When *R* is evaluated as 70% of *OMV*, the method can detect fewer incorrect and miss fewer correct logo transitions, thus the performances are overall better.

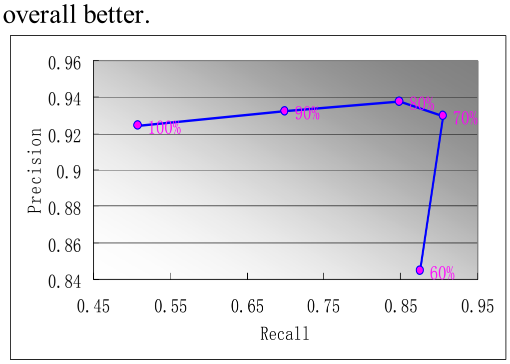

Fig. 7. The *P-R* curve of *R* value.

In the experiment, with random marked *C1* and *C2* positions in sports video, the proposed method has consistent performance, which shows that the method is stable to the position of logo transition in sports videos. Table 1 shows the final replay detection result on the test video database. The proposed method can extract the logo template sequence accurately. The logo template sequence matching method can reduce noise effectively during logo transition detection with logo template sequence method. Thus, the overall precision and recall performance of the method is high. In the experiment, some false and missing detection are partly due to logo transitions themselves missing and aberrance.

Table 1. The final replay detection result.

#Total replays	#Detect replays	#Correct replays	Recall	Precision
336	325	321	0.955	0.987

5. Conclusions

In this paper, a novel logo template sequence extraction algorithm has been proposed, and it can extract logo template sequence accurately. Because the logo template sequence matching can reduce noise effectively during logo transition detection, the algorithm can be used to detect replay segments in sports video efficiently. High performances of precision and recall make the experimental results very acceptable. The performance proves that the proposed replay segment detection method is very generic to common sports videos. The logo template sequence extraction algorithm is semi-automatic and the replays are verified via their lengths. Thus, extracting logo template sequence automatically and analysis of video shot to verify replay are promiseful method.

6. Acknowledgements

The authors would like to thank Dr. Wei Zeng and other friends in Multi-media Group, NEC Laboratories China, for their help and useful discussions during this work.

7. References

[1] V. Kobla, D. DeMenthon, D. Doermann, "Detection of slow-motion replay sequences for identifying sports videos", Proceedings of IEEE Third Workshop on Multimedia Signal Processing, pp.135-140, 1999.

[2] A. Ekin, A. M. Tekalp, "Automatic soccer video analysis and summarization", Symp. Electronic Imaging: Science and Technology: Storage and

Retrieval for Image and Video Databases IV, Jan.2003.

[3] X. Tong, Q. Liu, "Highlight ranking for sports video browsing," Proc. of ACM Multimedia 05.

[4] H. Pan, P. Beek, M. Sezan, "Detection of slow-motion replay segments in sports video for highlights generation", ICASSP2001, May 2001.

[5] H. Pan, B. Li, and M. Sezan, "Automatic detection of replay segments in broadcast sports programs by detection of logos in scene transitions", Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), 2002.

[6] Xiaofeng Tong, Hanqing Lu, Qingshan Liu, Hongliang Jin, "Replay detection in broadcasting sports video", Proceedings of the Third International Conference on Image and Graphics. 2004.

[7] X.Tong, Q.Liu, "Shot classification in sports video", Proceedings of ICSP 04, pp.1364-1367

[8] Lei Wang, Xu Liu, Steve Lin, Guangyou Xu, Heung-Yeung Shum, "Generic Slow-motion Replay Detection in Sports Video", IEEE International Conference on Image Processing (ICIP'04), Singapore, 2004

[9] Y.Deng, B.S.Manjunath, "Unsupervised Segmentation of Color-Texture Regions in Images and video", IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI '01), August 2001.

[10] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation", IEEE Trans. Patt. Anal. Mach. Intell, 24, 2002.